KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Computer Science Engineering

Course Activity Report

Database Management System

Submitted in the partial fulfillment for the academic requirement of

4th Semester B.E.

TITLE: Airline Reservation System

Submitted by

SL.No.	Batch Members Names	USN
1.	Vaibhav Muchandi	2GI18CS172
2.	Varun Shiri	2GI18CS175
3	Vijith Poojary	2GI18CS176
4	Vinayak Majati	2GI18CS177

2019 - 2020

Table of Contents

	Batch No: PG-04					
1.	Project Title: Airline Reservation System	Marks Range	USN (2GI18CS)			
			172	175	176	177
2.	Abstract (PO2)	0-2				
3.	Application of the topic to the course (PO2)	0-3				
4.	Literature survey and its findings (PO2)	0-4				
5.	Methodology, Results and Conclusion (PO1, PO3, PO4)	0-6				
6.	Report and Oral presentation skill (PO9, PO10)	0-5				
	Total	20				

Chapter 1

Introduction:

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

Characteristics of Database Management System:

1. Provides security and removes redundancy
2. Self-describing nature of a database system
3. Insulation between programs and data abstraction
4. Support of multiple views of the data
5. Sharing of data and multiuser transaction processing
6. DBMS allows entities and relations among them to form tables.
7. It follows the ACID concept (Atomicity, Consistency, Isolation, and Durability).
8. DBMS supports multi-user environment that allows users to access and manipulate data in parallel.

Databases touch all aspects of our lives. Some of the major areas of application are as follows:

- Banking
- Airlines
- Universities
- Manufacturing and selling
- Human resources

Chapter 2

Literature Survey:

NoSQL stands for “Not Only SQL” is an alternative to the relational database. In the relational database model, tables are used to store data. But the NoSQL has a variety of data models, like key-value pairs and graph formats. One of the most flexible ways of these models is the key-value pair. MongoDB is one popular NoSQL database that stores data in BSON. BSON is binary encoding JSON which stores data in key-value pairs.

Working with MongoDB NoSQL database is much easier than working with any relational database. There are no tables in MongoDB. All the data is stored in JSON format, that is key-value pairs. In JSON, we define a unique key with a value associated with it. These key-value pairs are stored in a document, which in turn is stored in a collection. A collection in MongoDB can have any number of documents and such documents can have any number of key-value pairs. As mentioned earlier, data in the MongoDB database is stored in BSON. BSON is nothing but extended JSON. It supports more data types than JSON. We store anything like, string, integer, Boolean, double, binary data, object, array, JavaScript code and many more.
These documents are grouped inside a collection. A collection can be equivalent to a table in a relational SQL database. A collection always exists in a database and there is no pre-defined structure of a collection. In SQL, the database contains tables and in MongoDB, the database contains collections.

Features of MongoDB:

1. **Indexing:** Fields in a MongoDB document can be indexed with primary and secondary indices.
2. **Replication:**
   MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data. Each replica-set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy

of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

3. **Ad-hoc queries:** MongoDB supports field, range query, and regular-expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.
4. **Server-side JavaScript Execution:** JavaScript can be used in queries, aggregation functions (such as MapReduce), and sent directly to the database to be executed.

Advantages of MongoDB:

1. MongoDB is a NoSQL database. We do not need to design the schema of the database when we are using MongoDB. Thus, the code we write defines the schema.
2. No complex joins are needed in MongoDB. There is no relationship among data in MongoDB.
3. The document query language supported by MongoDB is very simple as compared to SQL queries.
4. It is very easy to store arrays and objects since MongoDB uses JSON format to store data.
5. There is no need for mapping of application objects to database objects in MongoDB.

Disadvantages of MongoDB:

1. MongoDB uses high memory for data storage.
2. There is a limit for document size.
3. There is no transaction support in MongoDB.

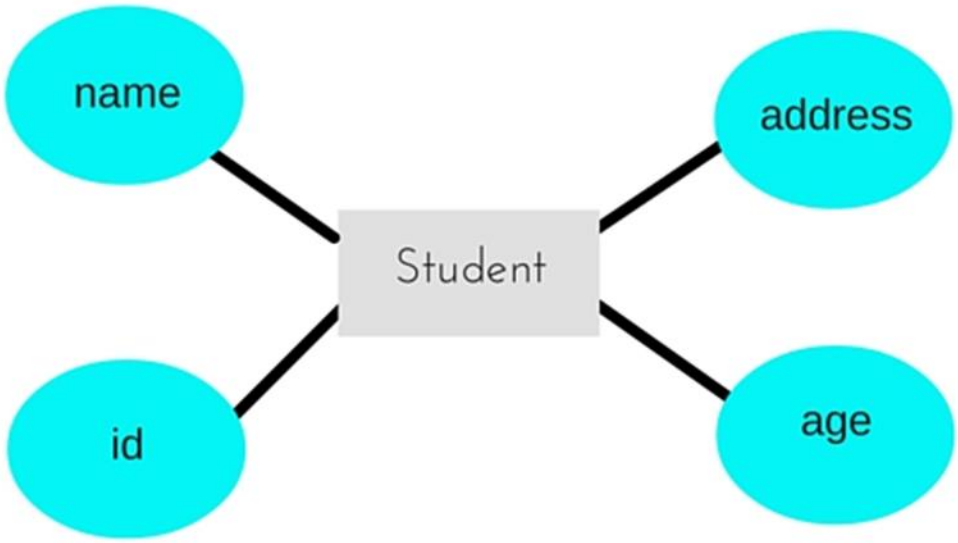
Chapter 3

Database Design:

A Database model defines the logical design and structure of a database and defines how data will be stored, accessed and updated in a database management system. While the Relational Model is the most widely used database model, there are other models too:

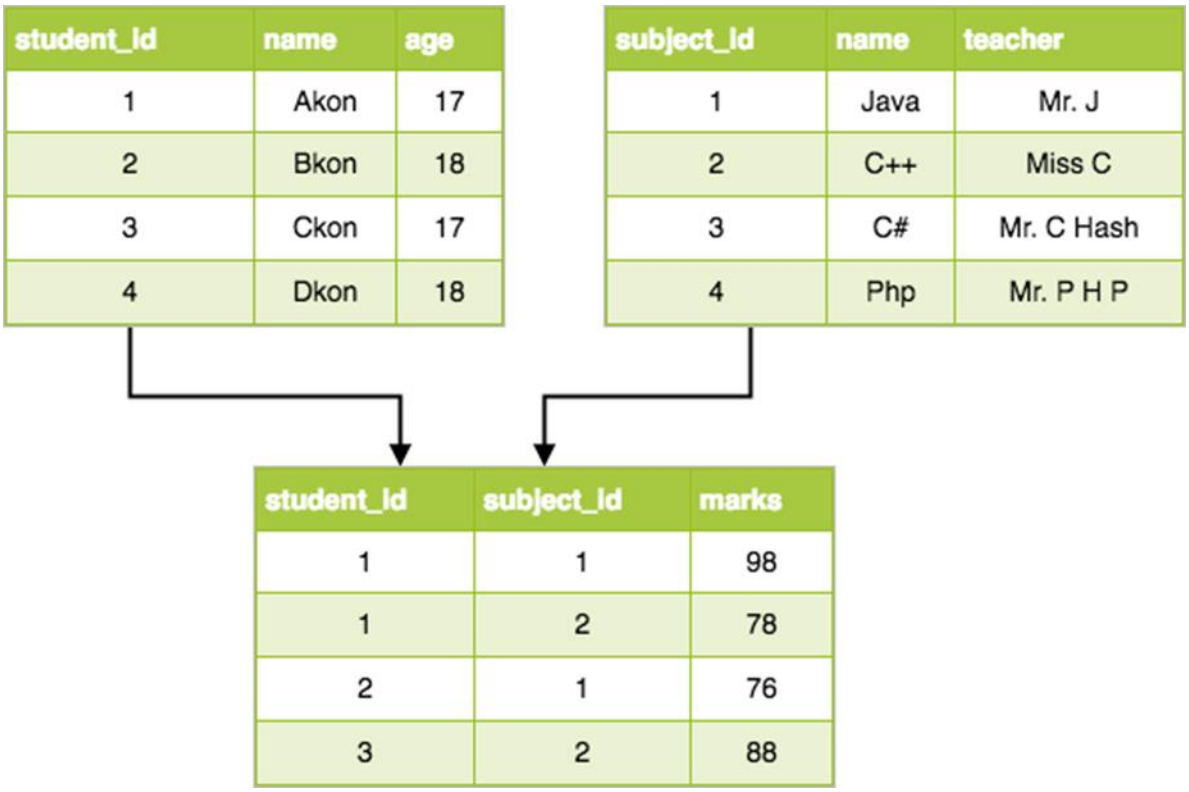
- Hierarchical Model
- Network Model
- Entity-Relationship Model
- Relational Model

Entity – Relationship Model:
In this database model, relationships are created by dividing object of interest into entity and its characteristics into attributes.



Relational Model:
In this model, data is organised in two-dimensional tables and the relationship is maintained by storing a common field.

The basic structure of data in the relational model is tables. All the information related to a particular type is stored in rows of that table. Hence, tables are also known as relations in relational model.



student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

After viewing the concept of the model, some of these are the advantages of Data Modelling:

- Provide very efficient “High Speed” retrieval.
- Ability to handle more relationship types.
- Ease of data access
- Data Integrity
- Data Independence

The Design Process:

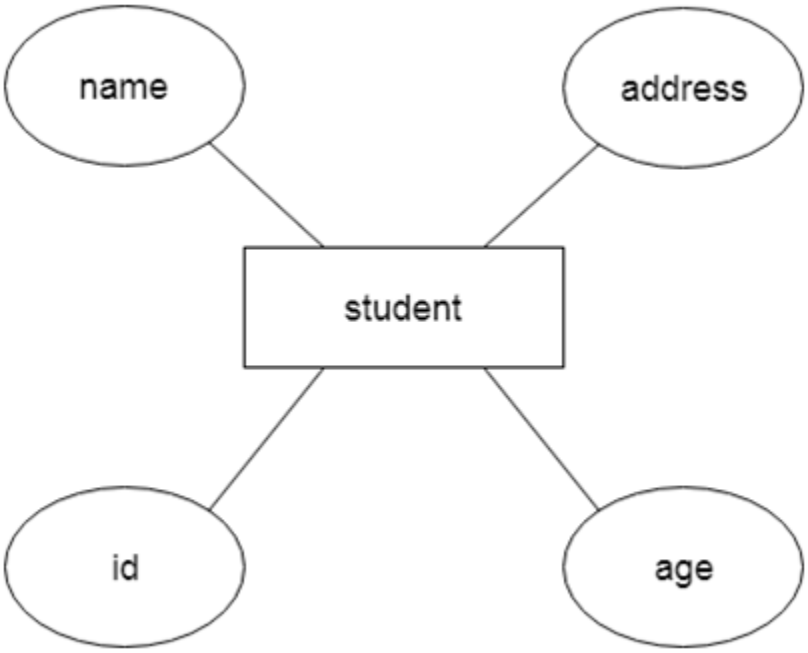
The Design Process consists of the following steps:

- **Determine the purpose of your database**
  - This helps prepare you for the remaining steps
- **Find and Organize the Information required**
  - Gather all of the types of information you might want to record in the database, such as product name and order number.
- **Divide the information into tables**
  - Divide your information items into major entities or subjects, such as Products or Orders. Each subject then becomes a table.
- **Turn information items into columns**
  - Decide what information you want to store in each table. Each item becomes a field, and is displayed as a column in the table. For example, an Employees table might include fields such as Last Name and Hire Date.
- **Specify primary keys**
  - Choose each table's primary key. The primary key is a column that is used to uniquely identify each row. An example might be Product ID or Order ID.
- **Set up the table relationships**
  - Look at each table and decide how the data in one table is related to the data in other tables. Add fields to tables or create new tables to clarify the relationships, as necessary.
- **Refine your design**
  - Analyze your design for errors. Create the tables and add a few records of sample data. See if you can get the results you want from your tables. Make adjustments to the design, as needed.
- **Apply the normalization rules**
  - Apply the data normalization rules to see if your tables are structured correctly. Make adjustments to the tables, as needed.

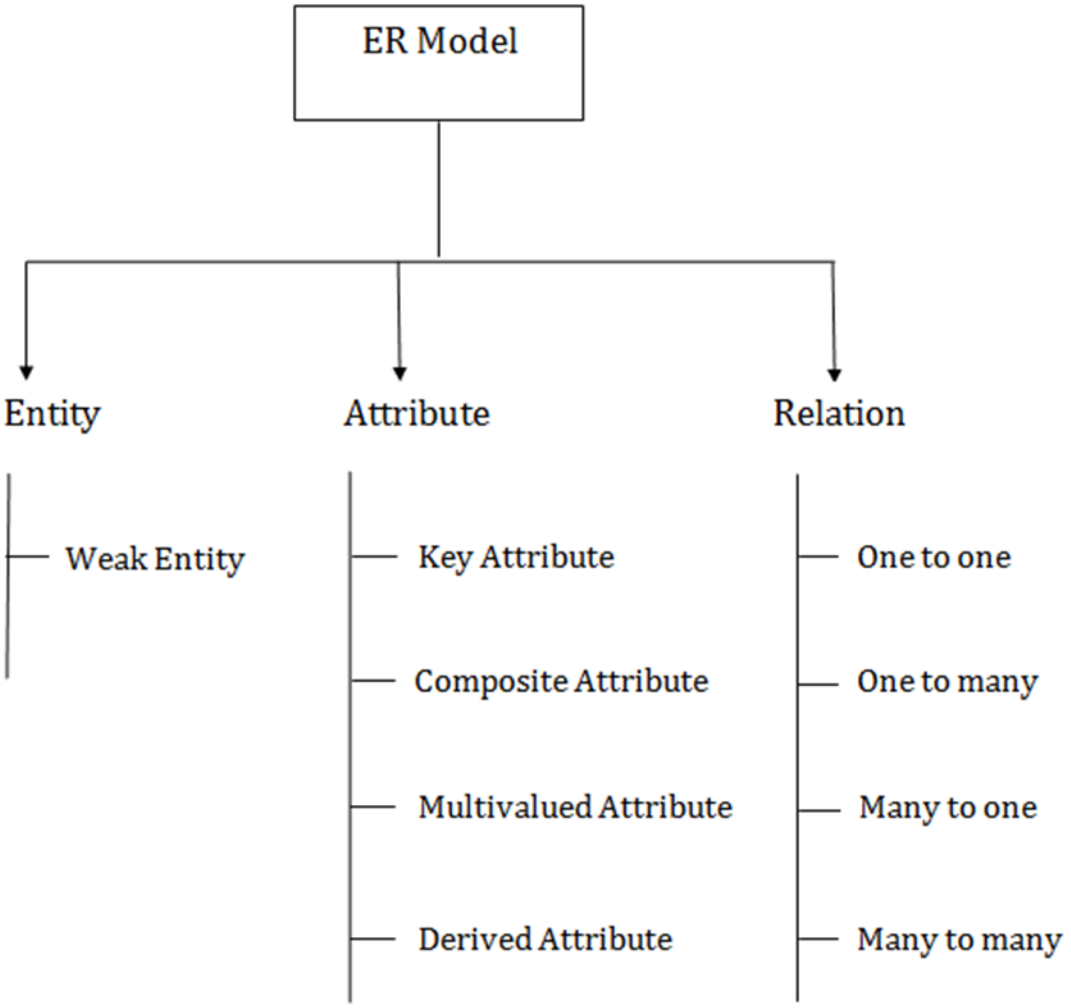
E – R Model:

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.

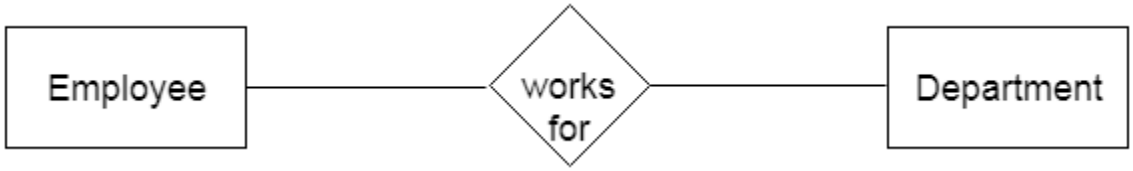
For example, suppose we design a school database. In this database, the student will be an entity with attributes like address, name, id, age, etc. The address can be another entity with attributes like city, street name, pin code, etc and there will be a relationship between them.



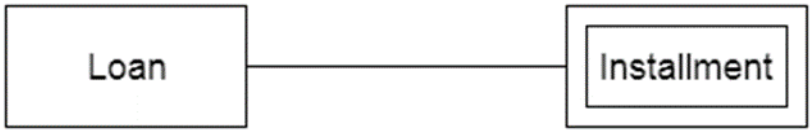
Components of E – R Diagram:



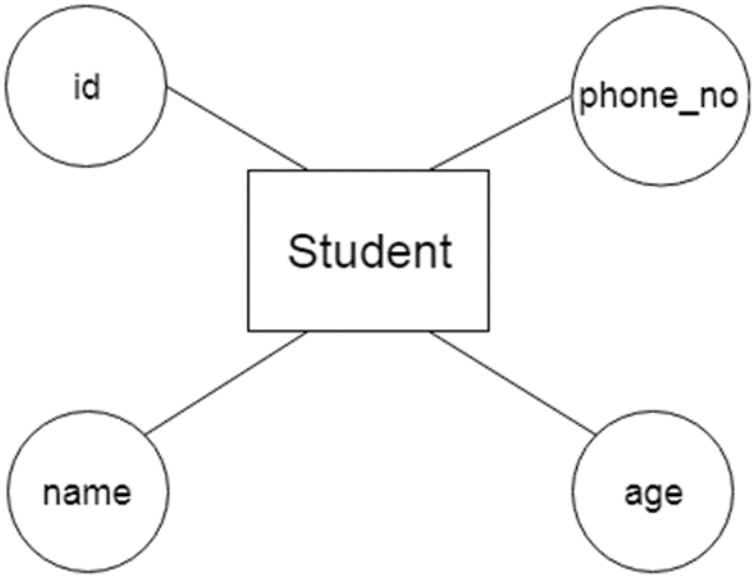
1. **Entity:**
   An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.
   Consider an organization as an example - manager, product, employee, department etc. can be taken as an entity.



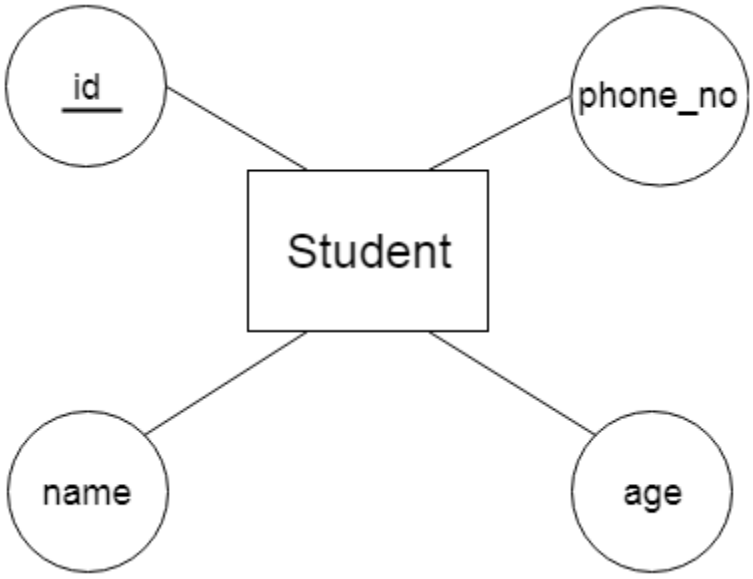
a. **Weak Entity:**
   An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.



2. **Attribute:**
   The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.
   For example, id, age, contact number, name, etc. can be attributes of a student.

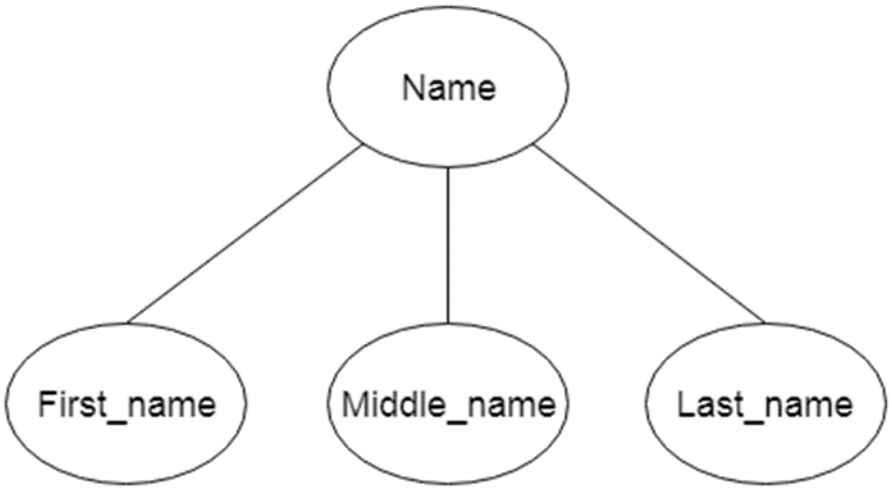


a. **Key Attribute:**
   The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.



b. Composite Attribute:

An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.



c. Multivalued Attribute:

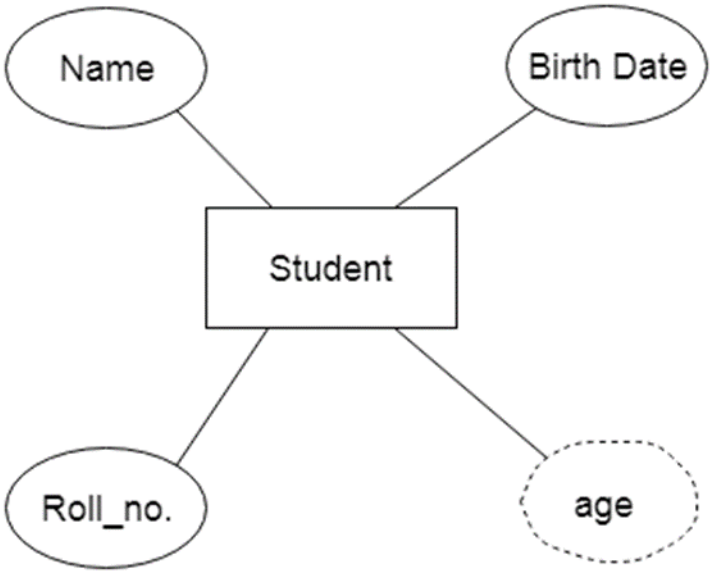
An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

For example, a student can have more than one phone number.

d. Derived Attribute:

An attribute that can be derived from another attribute is known as a derived attribute. It can be represented by a dashed ellipse.

For example, A person's age changes over time and can be derived from another attribute like Date of birth.



3. **Relationship:**

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

Types of relationship are as follows:

a. **One – to – One Relationship:**

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

For example, A female can marry to one male, and a male can marry to one female.

b. **One – to – Many Relationship:**

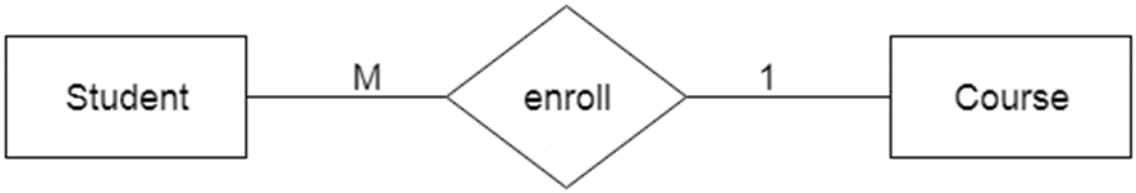
When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a a one-to-many relationship.

For example, Scientist can invent many inventions, but the invention is done by the only specific scientist.

c. Many – to – One Relationship:

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

For example, Student enrolls for only one course, but a course can have many students.



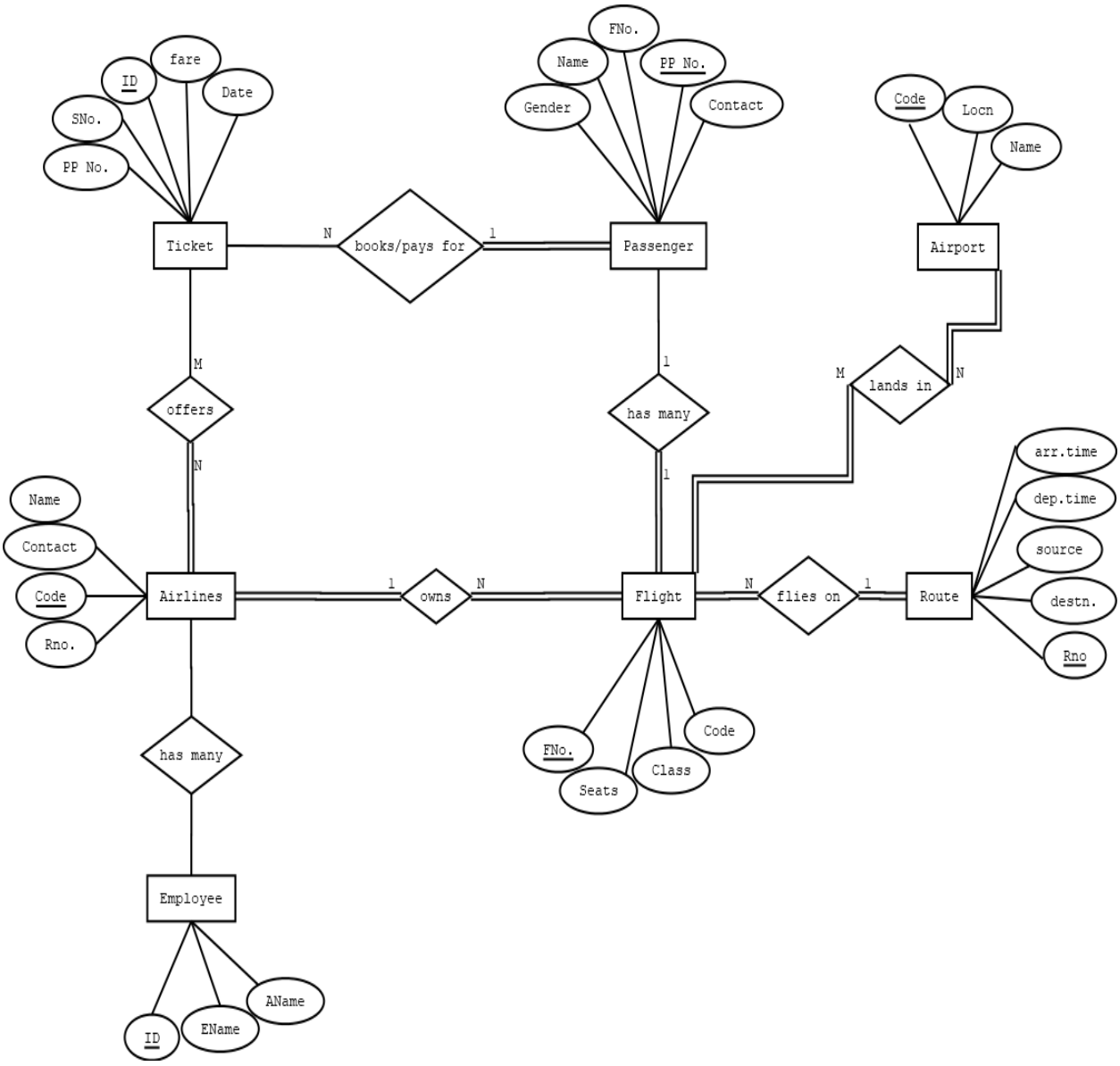
d. Many – to – Many Relationship:

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

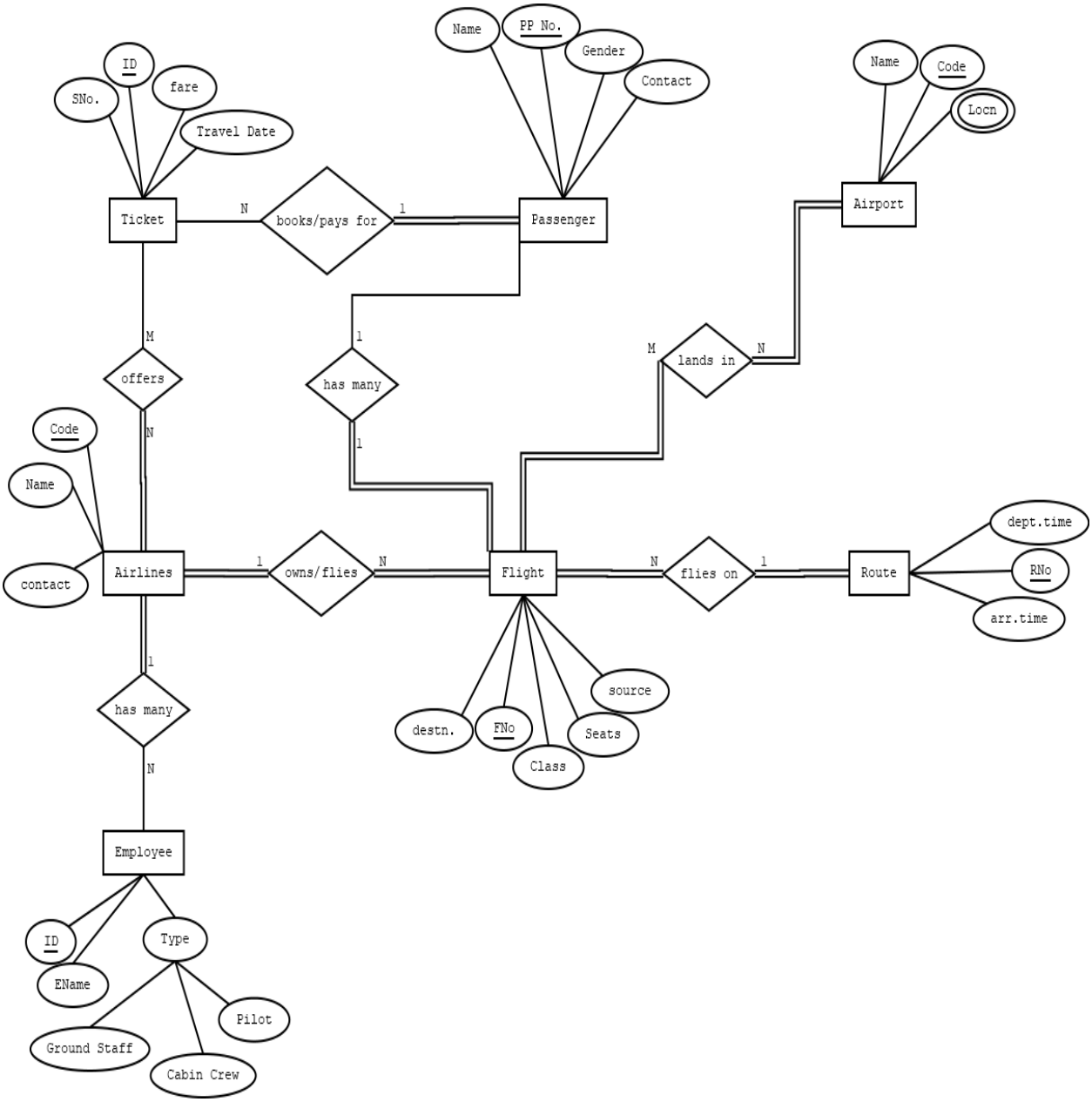
For example, Employee can assign by many projects and project can have many employees.

Airline Reservation System E – R Model:

Initial Design:



Rectified Design (with suggestions incorporated):



Justification:

The following are the reasons why the rectified design is better than the initial design:

- In the initial design, the foreign – key constraint is violated.
- The attributes are related to the wrong entities.
- Some more attributes are added to the entities in the rectified design.
- The participation constraints between the entities are corrected.

Relational Model:

RELATIONAL MODEL (RM) represents the database as a collection of relations. A relation is nothing but a table of values. Every row in the table represents a collection of related data values. These rows in the table denote a real-world entity or relationship.

Relational Model Concepts:

- **Attribute:** Each column in a Table. Attributes are the properties which define a relation. e.g., Student_Rollno, name, etc.
- **Tables:** In the Relational model the, relations are saved in the table format. It is stored along with its entities. A table has two properties rows and columns. Rows represent records and columns represent attributes.
- **Tuple:** It is nothing but a single row of a table, which contains a single record.
- **Relation Schema:** A relation schema represents the name of the relation with its attributes.
- **Degree:** The total number of attributes which in the relation is called the degree of the relation.
- **Cardinality:** Total number of rows present in the Table.
- **Column:** The column represents the set of values for a specific attribute.
- **Relation Instance:** Relation instance is a finite set of tuples in the RDBMS system. Relation instances never have duplicate tuples.
- **Relation Key:** Every row has one, two or multiple attributes, which is called relation key.
- **Attribute Domain:** Every attribute has some pre-defined value and scope which is known as attribute domain.

E – R Relational Mapping:

7 – Step Process:

1. Map Regular Entity Types
2. Map Weak Entity Types
3. Map Binary 1:1 Relation Types
4. Map Binary 1: N Relation Types

5. Map Binary M: N Relation Types
6. Map Multivalued Attributes
7. Map N – ary Relationship Types

Step 1: Map Regular Entity Types

- For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.
- Choose one of the key attributes of E as the primary key for R.
- If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Step 2: Map Weak Entity Types

- For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
- Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
- The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

Step 3: Map Binary 1:1 Relation Types

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.
- Choose one of the relations-say S-and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

Step 4: Map Binary 1: N Relation Types

- For each regular binary 1: N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.

- Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
- Include any simple attributes of the 1: N relation type as attributes of S.

Step 5: Map Binary M: N Relation Types

- For each regular binary M: N relationship type R, create a new relation S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.
- Also include any simple attributes of the M: N relationship type (or simple components of composite attributes) as attributes of S.

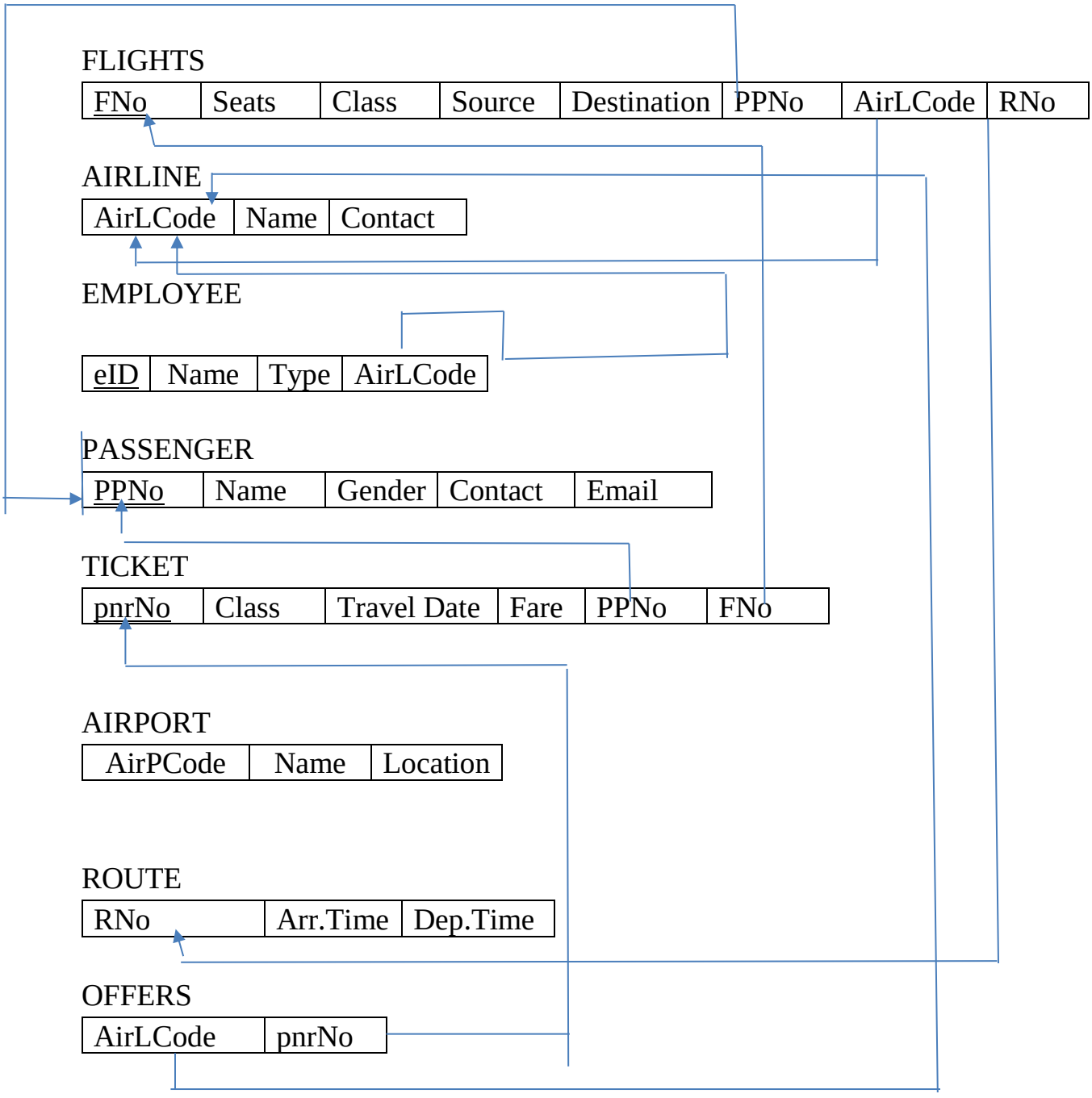
Step 6: Map Multivalued Attributes

- For each multivalued attribute A, create a new relation R.
- This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
- The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Step 7: Map N – ary Relation Types

- For each n-ary relationship type R, where n>2, create a new relationship S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
- Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

Relational Schema Diagram:



Chapter 4

Database Normalization:

NORMALIZATION is a database design technique that organizes tables in a manner that reduces redundancy and dependency of data. Normalization divides larger tables into smaller tables and links them using relationships. The purpose of Normalization is to eliminate redundant (useless) data and ensure data is stored logically.

First Normal Form (1NF):
In First Normal Form,

- Each table cell should contain a single value.
- Each record needs to be unique.

Second Normal Form (2NF):
There are two rules of Second Normal Form, which are:

- Rule 1: Be in 1NF
- Rule 2: Single Column Primary Key or there should be no Partial Dependency

Partial Dependency: When an attribute in a table depends on only a part of the primary key and not on the whole key.

Third Normal Form (3NF):
The rules of Third Normal Form are:

- Rule 1: Be in 2NF
- Rule 2: There should be no transitive functional dependencies

Transitive Dependency: When a non-prime attribute depends on other non-prime attributes rather than depending upon the prime attributes or primary key.

Boyce and Codd Normal Form (BCNF):

It is a higher version of the Third Normal Form. This form deals with certain type of anomaly that is no handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF.

- When a relation has more than one candidate key, anomalies may result even though the relation is in 3NF.
- 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keys that is, composite candidate keys with at least one attribute in common.
- BCNF is based on the concept of a determinant.
- A determinant is any attribute (simple or composite) on which some other attribute is fully functionally dependent.
- A relation is in BCNF if and only if every determinant is a candidate key.

Normalisation of our relations:

FLIGHTS

FNo	Seats	Class	Source	Destination	PPNo	AirLCode	RNo

In the Flights relation, the functional dependency is FNo.

The Seats attribute is a multivalued attribute. This relation can be converted into 1NF by splitting the Seats as individual tuples.

After converting to 1NF, the table is in 2NF and 3NF.

AIRLINE

AirLCode	Name	Contact

The above relation is in 2NF and 3NF as there is no partial or transitive dependency. The functional dependency is AirLCode.

EMPLOYEE

eID	Name	Type	AirLCode

The above relation is in 2NF and 3NF as there is no partial or transitive dependency. The functional dependency is eID.

PASSENGER

PPNo	Name	Gender	Contact	Email

The above relation is in 2NF and 3NF as there is no partial or transitive dependency. The functional dependency is PPNo.

TICKET

pnrNo	Class	Travel Date	Fare	PPNo	FNo

The above relation is in 2NF. The functional dependency is tID. However it is not in 3NF, as Fare depends on Travel Date and Class. Hence there is a transitive dependency. This can be rectified by splitting the table as follows:

TICKET

pnrNo	F_id	PPNo	FNo

DATE_FARE

F_id	Travel Date	Fare	Class

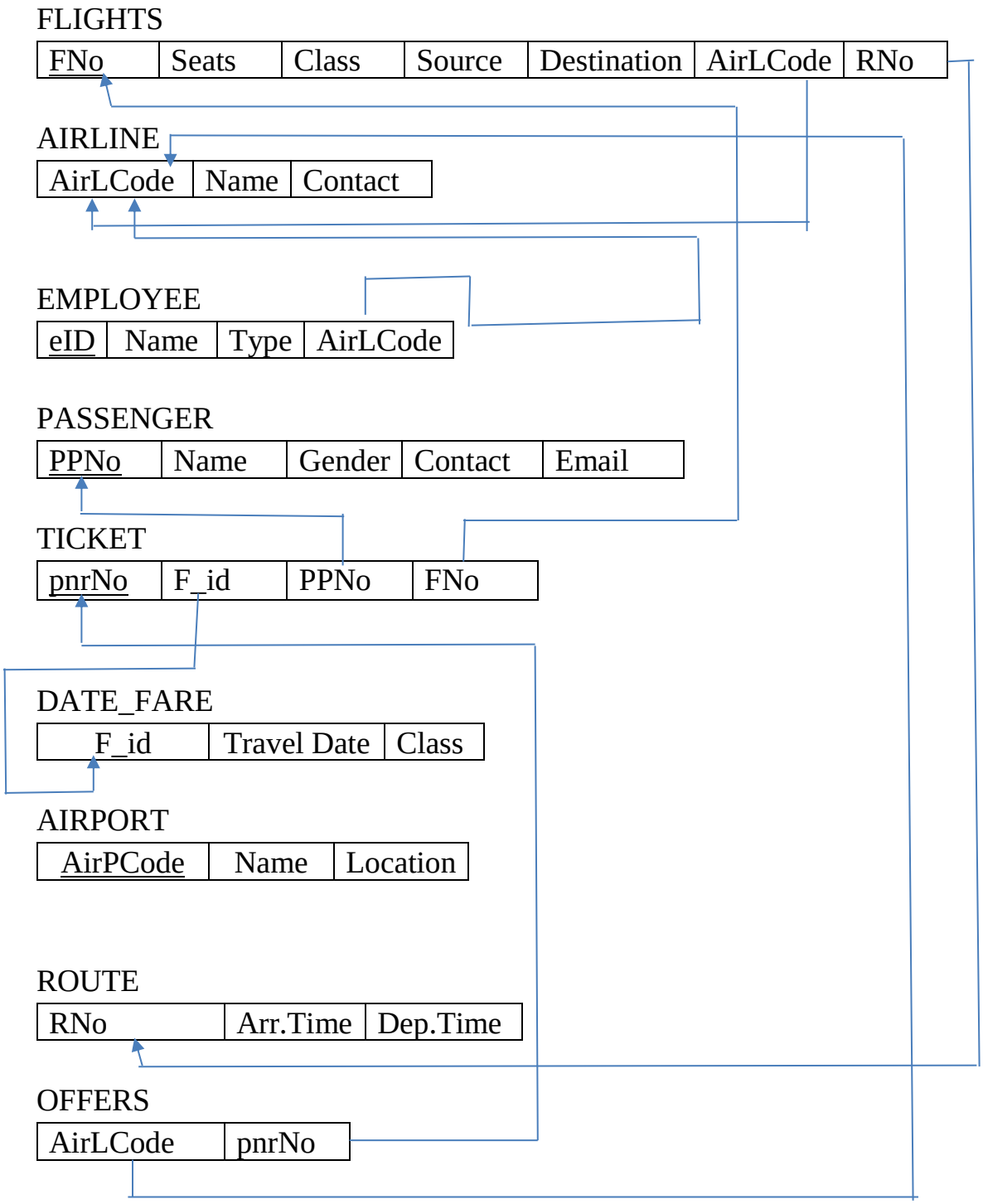
Now the table is in 3NF.

AIRPORT

AirPCode	Name	Location

The above relation is in 2NF and 3NF as there is no partial or transitive dependency. The functional dependency is AirPcode.

Final Schema Diagram



CREATE TABLE STATEMENTS

```
create table FLIGHTS(
        FNo varchar(10) PRIMARY KEY,
        Seats varchar(4),
        Class varchar(10),
        Source varchar(20),
        Destination varchar(20),
        AirLCode varchar(8) REFERENCES airline(AirLCode),
        RNo varchar(10) REFERENCES route(RNo)
);

create table AIRLINE(
        AirLCode varchar(8) PRIMARY KEY,
        Name varchar(20),
        Contact varchar(12)
);

create table EMPLOYEE(
        eID varchar(6) PRIMARY KEY,
        Name varchar(20),
        Type varchar(10),
        AirLCode varchar(8) REFERENCES airline(AirLCode);
);

create table PASSENGER(
        PPNo varchar(10) PRIMARY KEY,
        Name varchar(20),
        Gender varchar(1),
        Contact varchar(12),
        Email varchar(20)
);
```

```
create table TICKET(
        pnrNo varchar(10) PRIMARY KEY,
        F_id int REFERENCES date_fare(f_id),
        PPNo varchar(10) REFERENCES passenger(PPNo),
        FNo varchar(10) REFERENCES flights(FNo)
);

create table DATE_FARE(
        F_id int PRIMARY KEY,
        Travel_Date date,
        Fare dec(7, 2)
);

create table AIRPORT(
        AirPCode varchar(8) PRIMARY KEY,
        Name varchar(20),
        Location varchar(20)
);
```

Chapter 5

Conclusion:

Data Modelling is the method of developing the data model for the data to be stored in the database. A logical data model is to define the structure of the data elements and set the relationship between them. Finally, the physical model is used to specify the database – centric implementation of the model. Through the DIA tool, we were able to easily and efficiently construct the ER diagram for our database.

Through the process of Database Normalization, we bring our schema's tables into conformance with progressive normal forms. As a result, our tables each represent a single entity (a book, an author, a subject, etc.) and we benefit from decreased redundancy, fewer anomalies and improved efficiency.

References:

1. Elmasri and Navathe: Fundamentals of Database Systems, Addison – Wesley, 3 rd edition and onwards.
2. www.geeksforgeeks.org
3. www.guru99.com